



Uso de modelos de integração para transformações de modelos baseados em metamodelos

Ítalo de Pontes Oliveira¹, Teles de Sales Bezerra¹, Lucas Araújo Ramos¹, Rhavy Maia Guedes²

¹Graduandos em Tecnologia em Telemática - IFPB. e-mail: {italoifpb, teles.tux, lucas.araujo.ifpb}.gmail.com

²Professor Efetivo do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB. e-mail: rhavy.maia@gmail.com

Resumo: A engenharia de software oferece técnicas claras e objetivas aplicadas na construção de sistemas baseados em software, a fim de se obter melhores resultados no desenvolvimento, recomenda-se a divisão das etapas de projeto em: levantamento e análise de requisitos, modelagem de sistemas, codificação e testes antes da entrega do produto. O processo de modelagem é importante para permitir a criação de um sistema maduro, e com o processo de codificação mais eficiente, nela se inclui a transformação automática de modelo, uma técnica utilizada para geração de novos modelos baseados em outros já existentes, isso permite reduzir o tempo investido no processo de modelagem. Nesse contexto, foram avaliados os ganhos do uso de modelos de integração nas transformações, um método mais eficiente de se obter modelos a partir das transformações, os seguintes resultados foram obtidos: grande redução no número total de transformações entre modelos, maior organização no processo de modelagem, e redução no tempo e custo de desenvolvimento dos modelos.

Palavras-chave: engenharia de software, metamodelos, modelos de integração, transformação de modelos

1. INTRODUÇÃO

A engenharia de software oferece técnicas claras e objetivas na construção de sistemas, a fim de ser obtido melhores resultados no desenvolvimento, que é geralmente executado em fases de características específicas, para que seja proporcionado melhor legibilidade do sistema a ser desenvolvido, são divididas em: levantamento e análise de requisitos, modelagem do sistema, desenvolvimento e testes (Pressman, 2011). A análise de requisitos é o processo de avaliação das funcionalidades que o sistema possuirá, essa etapa evita redundância e esclarece os procedimentos a serem realizados pelo software. Já a modelagem do sistema monta uma visualização prévia das funcionalidades de abrangência. O desenvolvimento do sistema é envolvido pela definição do sistema quanto a sua estrutura de código. Os testes tratam falhas e exceções do desenvolvimento.

Para Sommerville, existem duas formas de desenvolvimento de software predominante no mercado: o modelo genérico, e o customizado. O modelo genérico permite que organizações produzam sistemas para serem vendidos no comércio com propósito geral, onde vários clientes são capazes de comprá-lo, no entanto, o modelo customizado é definido para resolver um problema específico e adaptado a uma única realidade.

Desse modo, é necessário que os sistemas sejam bem compreendidos pelas equipes de desenvolvimento, através do uso de modelos que representam todos os requisitos a serem implementados, além de fornecer um documento de contrato entre o desenvolvedor e o cliente, permite que seja especificado as características do software (Qiuyan, 2011). Os modelos são representações gráficas que trazem consigo informações importantes sobre o sistema (OMG, 2011). Eles são definidos a partir de seus metamodelos, que contém uma sintaxe abstrata e semântica estática de uma determinada linguagem de modelagem específica de domínio (Liu, 2005).

Em (Zhang, 2008) é visto que o tipo de transformação do modelo pode ser classificado entre totalmente automático, parcialmente automático ou processamento manual. A construção de um modelo a partir de uma transformação parcialmente automática e manual, é feita em sua maioria pelo engenheiro de software, que é mais suscetível ao erro do que a máquina, por isso, é usado transformações automáticas para obter menor probabilidade de erro e maior garantia de compatibilidade de dados. Algumas informações sobre os modelos armazenadas nos metamodelos,

são redundantes, ou seja, estão presentes em diferentes modelos, e são estas informações compartilhadas são utilizadas na transformação, para isso, é necessário implementar regras que definam as características comuns, e que deverão ser obtidas para a transformação.

2. MATERIAIS E MÉTODOS

O material foi obtido em sua maioria, da biblioteca digital do IEEE Xplore, também foram usados para o embasamento teórico, livros bastante referenciados na engenharia de software, como Pressman e Sommerville. Entre outros documentos utilizados, disponibilizados pela Object Management Group (OMG), e outras entidades. O trabalho foi desenvolvido pelo grupo de Iniciação Científica denominado Grupo de Estudo e Pesquisa em Engenharia de Software (GEPES) que investiga temáticas relacionadas a Modelagem Dirigida a Modelos (MDD) no Instituto Federal de Educação, Ciência e Tecnologia da Paraíba campus Campina Grande. O grupo é formado por estudantes do curso Superior em Telemática e pelo professor orientador. Durante o período de revisão da literatura foram levantados aspectos importantes sobre a transformação automática de modelos como, por exemplo, transformações de modelos e a utilização de modelos de integração, abordados a seguir.

3. TRANSFORMAÇÃO DE MODELOS

Representada na Figura 1 a maneira convencional de se fazer transformação automática de modelos, os círculos representam os modelos a serem transformados, e as setas indicam a direção da transformação. A seta que liga o círculo A para o círculo B, indica que a partir do metamodelo de entrada A, é obtido o metamodelo de saída do modelo B. Uma ferramenta de transformação de modelos, pode fazer com que o metamodelo de origem, seja transformado em um metamodelo de saída, como em (Qiuyan, 2011), aonde é proposto uma transformação de modelos AADL para UML. É dado um exemplo, onde se verifica que a ferramenta tem capacidade de transformar modelos complexos, para isso, selecionaram um modelo experimental de sistema de aviação. O número de linhas do metamodelo AADL chega a 60000, após a transformação se obteve diagrama de classes, diagrama de objeto, diagrama de sequencia e diagrama de estado.

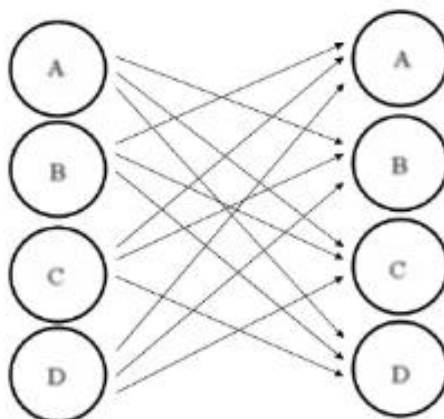


Figura 1 - Relação das transformações feitas dos modelos de entrada (esquerda) para os modelos de saída (direita).

Quando observado o método de transformação manual ou parcialmente automática, onde especificações são implementadas pelo engenheiro de software para transformações entre modelos do mesmo tipo, o número de transformações corresponde ao número de modelos ao expoente dois. No caso de transformação automática (não incluindo transformações de modelos entre si mesmo), o número de transformação corresponde a multiplicação do número de modelos pelo número de modelos menos um. Observado na Fórmula 1.

$$Nts = Nm * (Nm - 1)$$

Fórmula 1 - Fórmula para cálculo do número de transformações sem o uso do modelo de integração

Onde Nts é o número de transformações a serem feitas sem modelo de integração, e Nm o número de modelos de entrada. Esta fórmula é determinada para o exemplo da Figura 1, em que o modelo de entrada pode ser transformado para todos os outros modelos, menos a si mesmo.

4. MODELO DE INTEGRAÇÃO

Foi avaliado a proposta de usar Modelos de Integração, para deixar o método de transformação de modelos mais eficiente, o qual combina características comuns dos modelos a serem transformados. Os modelos de entrada, contém informações redundantes com os outros modelo, e informações específicas apenas para si, as especificações comuns a outros modelos, são informações que permite as transformações, e o modelo de integração corresponde a parte de qualquer intersecção entre modelos, mostrada na Figura 2, onde A, B, C e D são modelos com metamodelos conhecidos, e a área cinzenta entre eles, corresponde a informação associada com o modelo de integração.

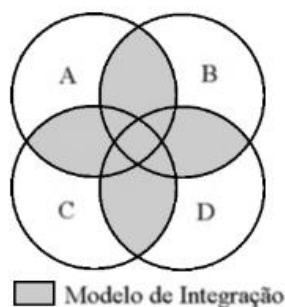


Figura 2 - Representação lógica da abrangência do Modelo de integração.

Com esta técnica, a representação para a transformação de modelos da Figura 1 sofre alterações, na Figura 3 é visto que as transformações não serão feitas diretamente entre cada modelo, é observado que o modelo de fonte é transformado para o modelo de integração, e do modelo de integração para o modelo de saída. Desta maneira, é possível transformar um modelo para si mesmo, só que desta forma ocorre perda de informação, pois, a informação será recuperada somente daquelas armazenadas no modelo de integração, que não corresponde a toda a informação contida no modelo de entrada. Isto não é visto como uma desvantagem, pois, não se faz uso de técnicas de transformação automática de modelos para transformação de modelos do mesmo tipo. Para modelos diferentes, sem a utilização do modelo de integração, só é possível recuperar as informações em comum do modelo de entrada com o modelo de saída, justamente, a mesma abrangência dada pelo modelo de integração, para este caso, não ocorre perda de dados.

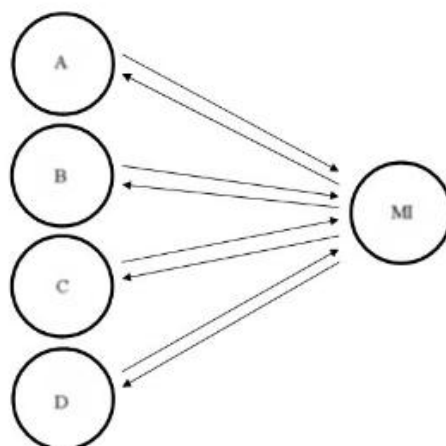


Figura 3 - Relação das transformações feitas dos modelos de entrada (esquerda) para o modelo de Integração - (MI, na direita) e depois para o modelo de saída (esquerda).

A sintaxe deste gráfico é o mesmo que o da Figura 1, os círculos representam os modelos, e as setas representam a direção das transformações. A partir a Figura 3, observa-se que o número de transformações corresponde a duas vezes o número de modelos de entrada, chegamos a seguinte fórmula para calcular o número de transformação de modelos com o uso de modelo de integração:

$$Ntc = 2 * Nm$$

Fórmula 2 - Fórmula para cálculo do número de transformações sem o uso do modelo de integração.

Onde Ntc é o número de transformações com modelo de integração, e Nm, o número de modelos de entrada.

5. TRABALHOS E DISCUSSÕES

A partir dos valores obtidos das Fórmulas 1 e 2, foi traçado um gráfico representado na Figura 4, utilizando estes valores para melhor visualização da relação do número de transformações.

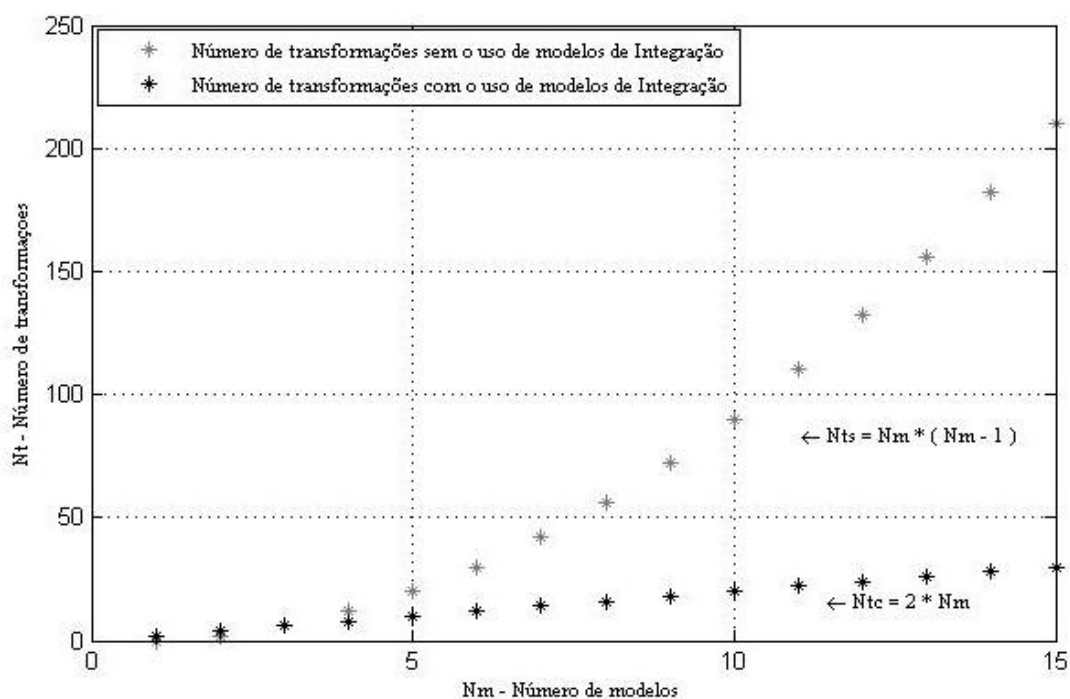


Figura 4 - Relação entre o número de modelos a serem transformados e o número de transformações.

Onde N_{ts} , é o número de transformações de modelos sem modelo de integração, e N_{tc} é o número transformações de modelos com modelos de integração. É observado que a transformação de um pequeno número de modelos de entrada (1, 2 e 3), que ao ser utilizado modelo de integração, não se oferece ganhos no número de transformações, e a partir disso, a proposta de utilização de modelos de integração, sempre tem proporcionado ganhos no número de transformações. Com os valores obtidos na Figura 4, foi construído uma relação entre o número de transformações sem MI e com MI, mostrado na Figura 5.

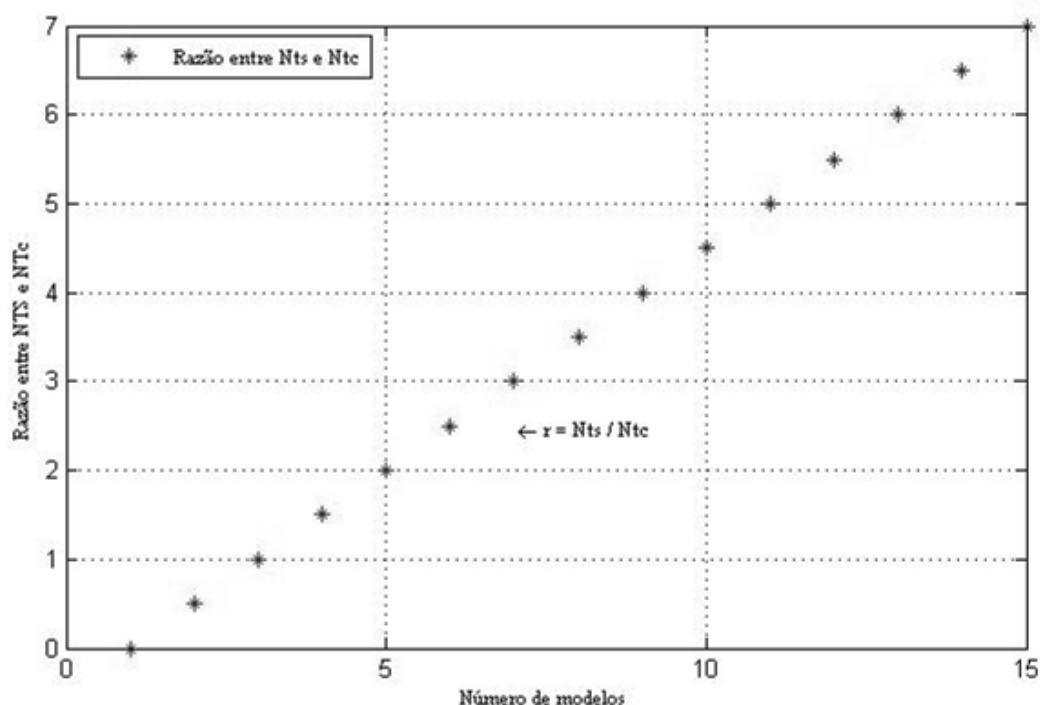


Figura 5 - Razão entre o número de transformações sem e com modelo de integração.s

Os pontos representam os valores de r , que é o quociente entre o número de transformações sem MI com o número de transformações com MI. Esta razão aumentada de $1/2$ em $1/2$, e pode ser calculada como o índice An de uma progressão aritmética (GIOVANNI, 2002), calculado na fórmula:

$$An = A1 + (n - 1) * r$$

Fórmula 3 - Cálculo para o índice An de uma progressão aritmética.

Onde An é a razão, $A1$ é o primeiro valor da razão (sempre é zero, ver Figura 5), n é o número de modelos para ser processados, e r a razão entre o Nts com Ntc , que foi definido anteriormente como $1/2$. Por conseguinte, a Fórmula 3 pode ser reescrita como:

$$An = (n - 1) * 0,5$$

Fórmula 4 - Fórmula do cálculo do índice An atualizada.

Ao ser usado o modelo de integração o ganho pode ser calculado, basta saber o número de modelos a serem transformados. Por exemplo, para 10 modelos a serem processados, substitui n por 10 na Fórmula 4, e obtém o valor de 4,5, que representa a razão entre o número de transformação com e sem modelo de integração. O número de transformações com o modelo de integração é igual ao dobro do número de modelos (vide Fórmula 2), sendo assim, obtém-se o número de 20 transformações, que multiplicado pela razão de 4,5 encontrada, chega-se ao valor de 90 transformações necessárias para fazer a mesma tarefa sem o modelo de integração.

Estes números mostram que para uma empresa de desenvolvimento de software que dedica tempo ao desenvolvimento de modelos, com apenas um modelo de entrada, conseguimos obter quantos modelos de saída forem necessários, precisa apenas uma pequena aplicação do engenheiro de software para complementar as informações não são cobertas pelo modelo de integração no modelo de saída.



5. CONCLUSÕES

Com o uso de modelos de integração para transformação de modelos, obtemos diversas vantagens, como: grande redução no número total de transformações entre modelos, maior organização no processo de modelagem, e redução no tempo e custo de desenvolvimento dos modelos. Além de melhorar a interoperabilidade entre os modelos desenvolvidos, permitido a partir de regras que ofereça a compatibilidade entre modelos. Nossa proposta para trabalhos futuros, é utilizar os conceitos a respeito do modelo de integração, para criar uma ferramenta comercial para transformações automática de modelos UML, visto que o ser humano é mais suscetível ao erro, esta ferramenta diminuirá a probabilidade de erro, e seguindo os conceitos defendidos pela engenharia de software, o processo de modelagem se tornará mais ágil.

REFERÊNCIAS

- DAN, L. QVT Based Model Transformation from Sequence Diagram to CSP, 2010. Trabalho apresentado ao: 15th IEEE International Conference on Engineering of Complex Computer Systems, Oxford, United Kingdom, 2010;
- GIOVANNI, José Ruy. Matemática complete: Ensino médio: Volume único / José Ruy Giovanni, José Roberto Bonjorno, José Ruy Giovanni Jr. – São Paulo: FTD, 2002;
- KIM, S.; CARRINGTON, D.; DUKE, R. A Metamodel-based transformation between UML and Object-Z. 2001. Trabalho apresentado ao: IEEE Human-Centric Computing Languages and Environments, Stresa: Italy, 2001;
- KÜSTER, J.M.; HECKEL, R.; ENGELS, G. Definition and Validating Transformations of UML Models, 2003. Trabalho apresentado ao: HCC 2003 - IEEE Symposium on Human Centric Computing Languages and Environments, Auckland: New Zealand, 2003;
- LEMESLE, R. Transformation Rules Based on Meta-Modeling, 1998. Trabalho apresentado ao: Enterprise Distributed Object Computing Workshop, 1998. EDOC '98. Proceedings. Second International, La Jolla, CA, 1998;
- LIU, J.; HE, K.; LI, C.; LIANG, P. A Transformation Definition Metamodel for Model Transformation, 2005. Trabalho apresentado ao: IEEE International Conference on Information Technology: Coding and Computing (ITCC'05), Las Vegas: United States, 2006;
- MARIBEL, Y. S.; RICARDO, J. M. On the Derivation of Class Diagrams from Use Cases and Logical Software Architectures, 2010. Trabalho apresentado ao: Fifth International Conference on Software Engineering Advances, Nice: France, 2010;
- METZGER, Andreas. A Systemic Look at Model Transformations, 2005. Trabalho apresentado ao: Springer, Model Driven Software development, Germany, 2005;
- MULLER, Joaquin,; MUKERJI, Jishnu. Model Driven Architecture (MDA): A Draft with annotations of issues to resolve Architecture Board ORMSC. In: Object Management Group, 2001. Disponível em: < http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf > Acesso em: 05 Ago 2012;
- OMG 2003b. MDA Guide Version 1.0.1, *Object Management Group*, OMG document omg/2003-06-01, 2003. Disponível em: < <http://www.omg.org/cgi-bin/doc?omg/03-06-01> > Acesso em: 05 Ago 2012;
- OMG. OMG Unified Modeling Language (OMG UML), Infrastructure. In: Object Management Group, 2011. Disponível em: < <http://www.sce.carleton.ca/courses/sysc-5708/f11/UML-infrastructure-2.3-formal-10-05-03.pdf> > Acesso em: 05 Ago 2012;
- PRESSMAN, Roger S. Software Engineering: a practitioner's approach / Roger S. Pressman – 5th ed. – New York, NY: McGraw-Hill, 2001.
- QIUYAN, L.; JIE, T.; QIUHONG, P.; JI, W.; CHAO, L. Automatic Transformation Technology from AADL Model to Model to UML Model, 2011. Trabalho apresentado ao:



Communication Software and Networks (ICCSN), IEEE 3rd International Conference on Communication Software and Networks (ICCSN 2011), Xi'an: China, 2011;

RAHIMI, S.K. Specification of UML Model Transformations, 2010. Trabalho apresentado ao: Third IEEE International Conference on Software Testing, Verification and Validation, Paris: France, 2010;

SAMUEL, A. A.; DORINA, P.; PANTANOWITZ, M. Using Model Transformation Semantics for Aspects Composition, 2010. Trabalho apresentado ao: IEEE Fourth International Conference on Semantic Computing, Pittsburgh: USA, 2010.

SHAN, L.; ZHU, H. Semantics of Metamodels in UML, 2009. Trabalho apresentado ao: Third IEEE International Symposium on Theoretical Aspects of Software Engineering, Tianjin: China, 2009;

SOMMERVILLE, Ian. Engenharia de Software / Ian Sommerville: tradução André Maurício de Andrade Ribeiro: revisão técnico Kechi Hirma. – São Paulo. Addison Wesley, 2003;

ZHANG, J.; LIU, S.; WANG, X.; QIN, T. A Model Transformation Classification Method Used in QVT. 2008. Trabalho apresentado ao: IEEE International Symposium on IT in Medicine and Education, Xiamen: China, 2008;